

REMARKS

The objection to the abstract has been corrected.

Claim 1 calls for writing blocks of a demultiplexed data frame at a first size into a register and reading blocks of a second data size different from the first size from the register.

The claim is rejected under Section 102 as being anticipated by Walker. However, in Walker, the blocks are never changed in size. While they may be read out on buses that are different sizes, the blocks are always the same size.

In Walker, a block is eight words. See paragraph 134, line 3. The block generator 302 forms blocks of eight words. Thus, plainly, the output from the generator 302, namely BLK, are blocks of eight words each.

The pre-coder 301 receives quads of words and their respective control flags and generates a code for each quad. See paragraph 132, lines 8-12. The quad type code indicates the block type of a block. The blocks are blocks of eight words. See paragraph 58, lines 5 and 6. In other words, what comes into the block generator 302 are blocks and what comes out of block generator 302 are blocks. There is nothing to indicate that the size of the blocks ever change from eight words. In other words, the item 301, before the item 302, works with blocks and blocks are defined to be eight words and the item 302 generates blocks which are also defined to be eight words. The only thing that the block generator 302 does is to separate out the codes from the actual blocks of data and to transmit the blocks on line 313 and the codes on the line 314. But the data is in the form of blocks of the same size, both before and after. Blocks are referred to throughout the patent application to Walker and there is nothing to indicate that blocks are ever anything other than eight words.

Whether those eight words are outputted or inputted through buses of different sizes, the blocks themselves never change their size. Thus, in Walker, the blocks are written and read in the same block size. Note that a quad is simply four blocks of eight words, each of eight bits. See paragraph 58.

After the codes are separated from the blocks, the codes are thereafter recombined with the blocks in the frame composer 305. See paragraph 135. Note the language in paragraph 135 “the pair of quad-type codes and the control word flags corresponding to the block are fed by the 18-bit wide bus 314 to the type word generator 306.” It is clear that after recombination, before

separation, and before storage in the register, the application specifically talks about blocks and never suggests that those blocks change size. The only thing that may change is the number of bits associated with the code words. Certainly, Walker talks about blocks and blocks are claimed, but Walker is reasonably clear that his blocks never change size. If the code words associated with those blocks were to change size that would be irrelevant to the claimed invention.

Other evidence of this is contained in the abstract. There, it is explained that blocks of input data are received and that these blocks comprise packets of information words and that those packets are preceded and followed by control words. Thus, it is plain that the frame never changes size. It is only the control words that vary. The office action does indicate that the cited reference converts 64-bit frames to 66-bit frames. It is not believed that this is correct. In fact, what it does is it takes quads of 8 bits or 64 bits of data, also called blocks, appends a master transition, and outputs the 66-bit frame. See paragraph 142. Thus, 64-bit frames are not converted to 66-bit frames. 64 bits of pure information words have the control information of two bits appended to actually form the frame. This is different from converting a 64-bit frame to a 66-bit frame.

At a higher level, at the point relied on by the Examiner, at block 302, no frame has yet been formed. All we have at that point is information words and control words. The information words and control information are separated out, processed, and the control information in the form of MT, on line 316 in Figure 8B, is provided to the frame assembler 308 which simply packs it onto the information words to form the frame.

The block size is never changed. They are always 64 bits and they are always 64-bit information words, which means they are free of control. No frame is ever formed except the 66-bit frame.

Thus, the cited reference does not receive a data frame of a first size. Nor is there any multiplexing blocks to form output data in a frame of a second size. In the cited reference, the frames are always 66 bits.

With respect to claim 11, the same distinction applies.

Claim 23 calls for examining a window to determine whether at least one synchronization bit is located within data within a window of a predetermined size within a stream of data. The window is shifted along the stream if a valid synchronization bit is not found in the window. But even if the Examiner were right that there is some shifting of the window, that shifting does not seem to be precipitated by whether or not a valid synchronization bit is found in the window. The

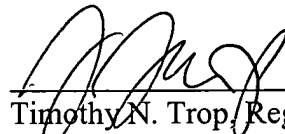
only synchronization bit that the Examiner appears to be asserting is the start of frame and the end of frame. But there is no indication that these elements are ever not found and that the absence of those bits results in shifting the window. Instead, it is indicated that the addition or deletion of fill words is used to maintain core synchronization with the local clock signal. See Mejia, column 7, lines 21-23.

Therefore, reconsideration would appear to be necessary.

On the same basis, reconsideration of claim 33 is requested.

Respectfully submitted,

Date: February 6, 2006



Timothy N. Trop, Reg. No. 28,994
TROP, PRUNER & HU, P.C.
8554 Katy Freeway, Ste. 100
Houston, TX 77024
713/468-8880 [Phone]
713/468-8883 [Fax]

Attorneys for Intel Corporation